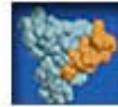
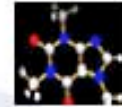
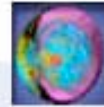




SciDAC

Scientific Discovery through Advanced Computing



SciDAC PDSI Update

HECIWG FSIO Workshop 2007, August 7, Arlington VA

Garth Gibson

Joint with Bianca Schroeder
Carnegie Mellon University

SciDAC Petascale Data Storage Institute (PDSI)

www.pdsi-scidac.org

w/ LANL (G. Grider), LBNL (W. Kramer), SNL (L. Ward),
ORNL (P. Roth), PNNL (E. Felix),
UCSC (D. Long), U.Mich (P. Honeyman)

Agenda

- SC0x Petascale Data Storage Workshops
 - CFP for SC07 workshop
- Highlights of PDSI recent progress
 - Call for static file systems stats collection
- Revisit Checkpoint/Restart in Petascale era
 - Is checkpoint/restart running out of steam

www.pdsi-scidac.org/pdsw06.html



Petascale Data Storage Workshop Supercomputing '06

Session Chair: Garth Gibson, CMU

Thurs Nov 16, 2006
[SC06 Panel Web Page](#)

PANEL: DOE HPC Practices and Problems: Today & Tomorrow

Panel Members:

- Bill Kramer, LBNL/NERSC – *NERSC Experience and Plans for Petascale Data*

[PDF \[2M\]](#)

- Evan Felix, PNNL – *PNNL EMSL Lustre Activities*

[PDF \[2.7M\]](#)

- Phil Roth, ORNL – *The Path to Petascale at Oak Ridge National Laboratory*

[PDF \[1.7M\]](#)

[Open Mic](#)

SESSION: Programming for Storage

Speaker:

- Dave O'Hallaron, CMU – *Best Practices in Programming for Effective Storage*

[PDF \[240K\]](#)

Speaker:

- Rob Ross, ANL – *HECE Posix Extensions for High End Computing*

[PDF \[380K\]](#)

[Open Mic](#)

PANEL: Training Programmers About Storage – What Needs to be Taught?

Remarks:

- Garth Gibson, CMU – *Programming for Storage*

[PDF \[325K\]](#)

SC07 Petascale Data Storage Workshop

- Full day workshop, Sunday November 11
- Refereed track for extended abstracts
 - 2-5 page extended abstracts
 - Program committee is PDSI co-PIs
 - Submission deadline Sept 28
 - Notification Oct 11
 - Selected papers & talks published on web site
- www.pdsi-scidac.org/SC07 (under construction)
- Everyone here strongly encouraged to participate

PDSI progress highlights

- Failure data gathering
 - LANL, NERSC & PNNL releases
 - J. Nunez and W. Kramer talks
 - Covering 4-9 years, HEC clusters, archive, storage devices
 - LANL data has seen 900 downloads in 6 months
 - Feeding papers into, at least, DSN and FAST
- Bianca Schroeder update tomorrow
 - Computer Failure Data Repository (CFDR) at USENIX now online at cfdi.usenix.org
 - Please contribute, use and cite
 - See also checkpoint/restart revisiting in this talk

PDSI progress highlights con't

- Virtualization
 - R. Farber's presentation
 - Xen-based intermingling of clients and servers to share resources but protect software
- Parallel NFS rapidly approaching RFC
 - NFSv4.1 definition and Linux impl. for files, blocks, objects
 - Inherently extensible for your new storage interface
- Tracing & characterization
 - L. Ward's instrumented Red Storm on sourceforge
 - sourceforge.net/projects/libsysio
 - www.pdsi-scidac.org/fsstats released for data collection
 - Static characterization of your file system

- Build public database of basic FS characteristics

```

----- CUT HERE ----- REPORT FOLLOWS ----- CUT HERE -----
Generated by fsstats v1.3 (04/05/2007)
complete : processed 999652 files in 343 secs (at 2914.44 files/
sec)
Skipped 797 duplicate hardlinked files
Encountered 142 errors while walking filesystem tree
RESULTS MAY BE INCOMPLETE
CSV WRITTEN TO garth-mac.csv
----- BEGIN NORMAL -----
total 87.48 GB used to store 84.85 GB user data, overhead 3.01%
file size:
count=841445 avg=105.68 KB
min=0.00 KB max=14395547.50 KB
[ 0- 2 KB): 249888 (29.70%) ( 29.70% cumulative)
[ 2- 4 KB): 191813 (22.80%) ( 52.49% cumulative)
[ 4- 8 KB): 152107 (18.08%) ( 70.57% cumulative)
[ 8- 16 KB): 87429 (10.39%) ( 80.96% cumulative)
[ 16- 32 KB): 55928 ( 6.65%) ( 87.61% cumulative)
[ 32- 64 KB): 37889 ( 4.50%) ( 92.11% cumulative)
[ 64- 128 KB): 17082 ( 2.03%) ( 94.14% cumulative)
[ 128- 256 KB): 12589 ( 1.50%) ( 95.64% cumulative)
[ 256- 512 KB): 14342 ( 1.70%) ( 97.34% cumulative)
[ 512- 1024 KB): 13006 ( 1.55%) ( 98.89% cumulative)
[ 1024- 2048 KB): 4497 ( 0.53%) ( 99.42% cumulative)
[ 2048- 4096 KB): 2329 ( 0.28%) ( 99.70% cumulative)
[ 4096- 8192 KB): 1712 ( 0.20%) ( 99.90% cumulative)
[ 8192- 16384 KB): 591 ( 0.07%) ( 99.97% cumulative)
[ 16384- 32768 KB): 147 ( 0.02%) ( 99.99% cumulative)
[ 32768- 65536 KB): 56 ( 0.01%) (100.00% cumulative)
[ 65536- 131072 KB): 17 ( 0.00%) (100.00% cumulative)
[ 131072- 262144 KB): 11 ( 0.00%) (100.00% cumulative)
[ 262144- 524288 KB): 6 ( 0.00%) (100.00% cumulative)
[ 524288- 1048576 KB): 3 ( 0.00%) (100.00% cumulative)
[ 1048576- 2097152 KB): 2 ( 0.00%) (100.00% cumulative)
[ 2097152- 4194304 KB): 1 ( 0.00%) (100.00% cumulative)
[ 4194304- 8388608 KB): 1 ( 0.00%) (100.00% cumulative)

capacity used:
count=841445 avg=108.88 KB
min=0.00 KB max=14395548.00 KB

```

```

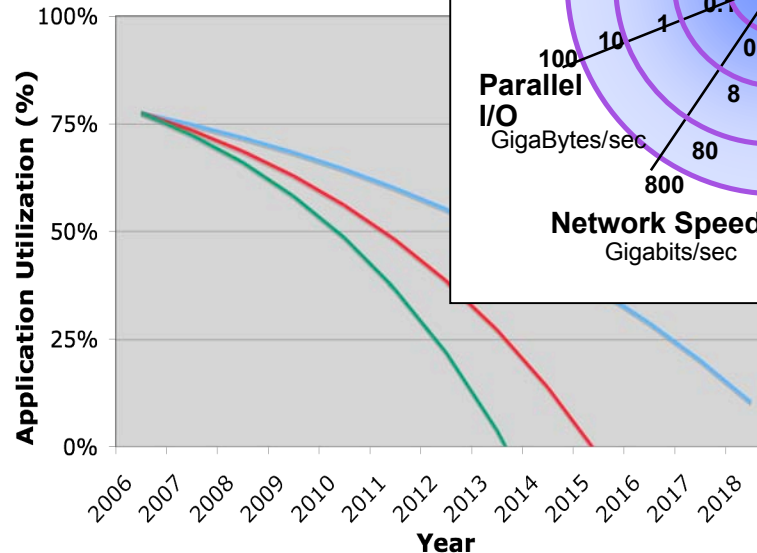
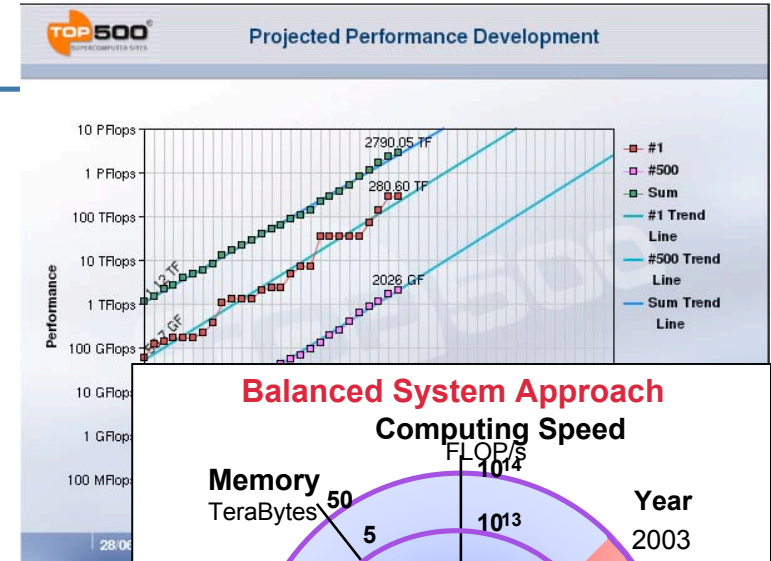
time since last modification (files):
count=841445 avg=623.64 days
min=-4054.03 days max=13652.14 days
[ < 0 days]: 1
[ 0- 2 days): 5484 ( 0.65%) ( 0.65% cumulative)
[ 2- 4 days): 3922 ( 0.47%) ( 1.12% cumulative)
[ 4- 8 days): 4666 ( 0.55%) ( 1.67% cumulative)
[ 8- 16 days): 8596 ( 1.02%) ( 2.69% cumulative)
[ 16- 32 days): 12023 ( 1.43%) ( 4.12% cumulative)
[ 32- 64 days): 47304 ( 5.62%) ( 9.74% cumulative)
[ 64- 128 days): 45467 ( 5.40%) ( 15.15% cumulative)
[ 128- 256 days): 94263 (11.20%) ( 26.35% cumulative)
[ 256- 512 days): 102212 (12.15%) ( 38.50% cumulative)
[ 512- 1024 days): 401080 (47.67%) ( 86.16% cumulative)
[ 1024- 2048 days): 102329 (12.16%) ( 98.32% cumulative)
[ 2048- 4096 days): 13318 ( 1.58%) ( 99.91% cumulative)
[ 4096- 8192 days): 210 ( 0.02%) ( 99.93% cumulative)
[ 8192-16384 days): 570 ( 0.07%) (100.00% cumulative)

time since last modification (kbytes):
count=88927979 avg=594.83 days
min=-4054.03 days max=13652.14 days
[ < 0 days): 3343
[ 0- 2 days): 19421900 (21.84%) ( 21.84% cumulative)
[ 2- 4 days): 122297 ( 0.14%) ( 21.98% cumulative)
[ 4- 8 days): 486548 ( 0.55%) ( 22.52% cumulative)
[ 8- 16 days): 834103 ( 0.94%) ( 23.46% cumulative)
[ 16- 32 days): 4819624 ( 5.42%) ( 28.88% cumulative)
[ 32- 64 days): 4251195 ( 4.78%) ( 33.66% cumulative)
[ 64- 128 days): 3716516 ( 4.18%) ( 37.84% cumulative)
[ 128- 256 days): 11162559 (12.55%) ( 50.39% cumulative)
[ 256- 512 days): 6123221 ( 6.89%) ( 57.28% cumulative)
[ 512- 1024 days): 18736649 (21.07%) ( 78.35% cumulative)
[ 1024- 2048 days): 16486111 (18.54%) ( 96.89% cumulative)
[ 2048- 4096 days): 2363439 ( 2.66%) ( 99.55% cumulative)
[ 4096- 8192 days): 9361 ( 0.01%) ( 99.56% cumulative)
[ 8192-16384 days): 391107 ( 0.44%) (100.00% cumulative)

```

Agenda II

- Scaling thru PetaFLOPS era
- Storage driven by coping with failure: checkpoint/restart
- Balanced systems model
 - Assumes constant MTTI
- But historical data says MTTI goes as # sockets
- Machine utilization for Hero Apps at risk
- Revisit checkpointing



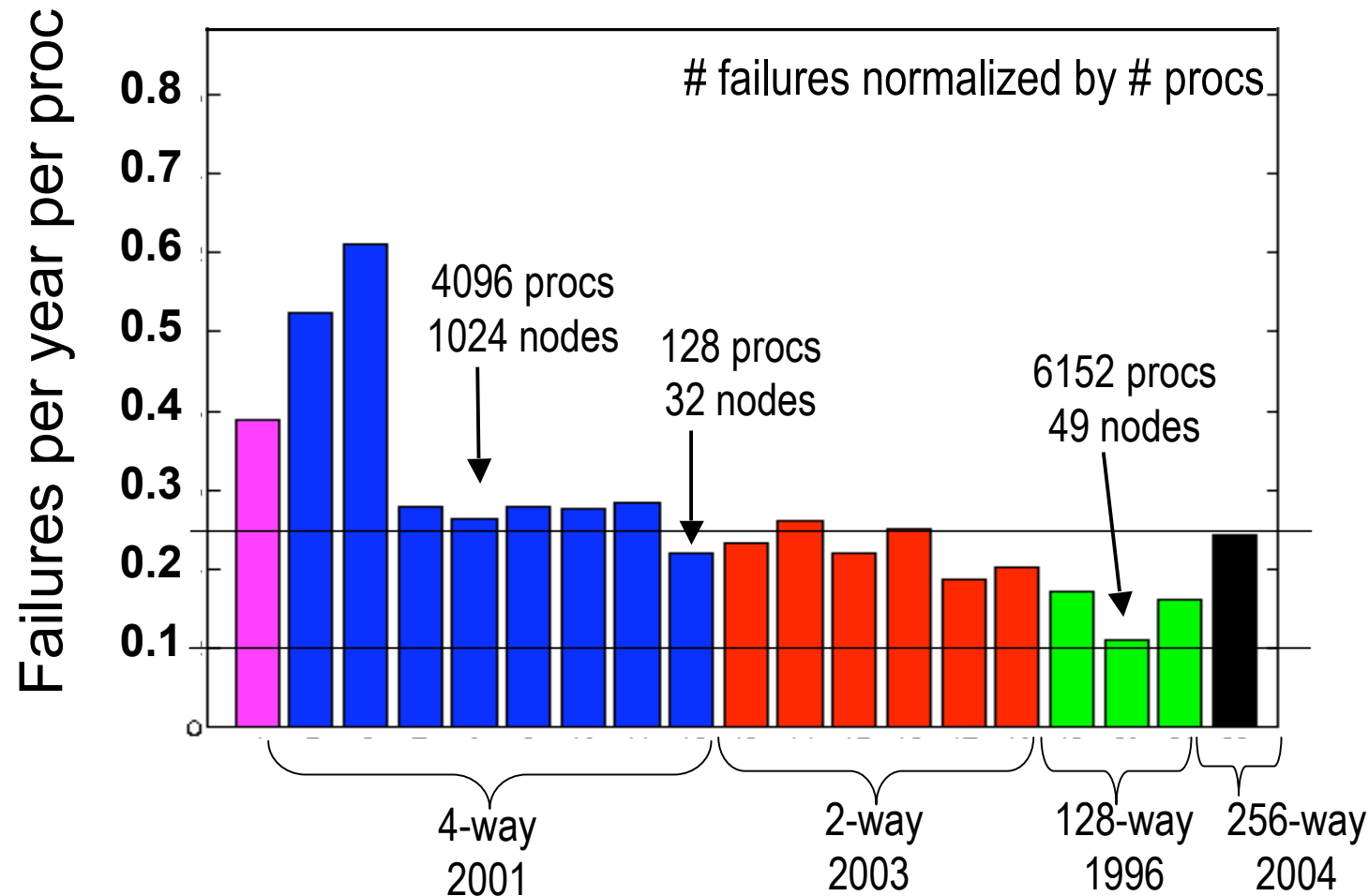
LANL interrupt history

- Los Alamos releases root cause logs for:
 - 23,000 events causing application interruption
 - 22 clusters & 5000 nodes
 - Covers 9 years & continues
- Kicks off our work understanding pressure on storage bandwidth
 - Checkpoint/restart
- More recent failure logs released from NERSC, PNNL, PSC, 2 anonymous

(I) High-level system information				(II) Information per node category			
HW	ID	Nodes	Procs	Procs /node	Production Time	Mem (GB)	NICs
A	1	1	8	8	N/A – 12/99	16	0
B	2	1	32	32	N/A – 12/03	8	1
C	3	1	4	4	N/A – 04/03	1	0
D	4	164	328	2	04/01 – now	1	1
				2	12/02 – now	1	1
	5	256	1024	4	12/01 – now	16	2
	6	128	512	4	09/01 – 01/02	16	2
	7	1024	4096	4	05/02 – now	8	2
				4	05/02 – now	16	2
				4	05/02 – now	32	2
				4	05/02 – now	352	2
	8	1024	4096	4	10/02 – now	8	2
				4	10/02 – now	16	2
				4	10/02 – now	32	2
				4	10/02 – now	32	2
	9	128	512	4	09/03 – now	4	1
	10	128	512	4	09/03 – now	4	1
	11	128	512	4	09/03 – now	4	1
	12	32	128	4	09/03 – now	4	1
				4	09/03 – now	16	1
E	13	128	256	2	09/03 – now	4	1
	14	256	512	2	09/03 – now	4	1
	15	256	512	2	09/03 – now	4	1
	16	256	512	2	09/03 – now	4	1
	17	256	512	2	09/03 – now	4	1
	18	512	1024	2	09/03 – now	4	1
F				2	03/05 – 06/05	4	1
	19	16	2048	128	12/96 – 09/02	32	4
				128	12/96 – 09/02	64	4
	20	49	6152	128	01/97 – now	128	12
				128	01/97 – 11/05	32	12
				80	06/05 – now	80	0
				128	10/98 – 12/04	128	4
	21	5	544	32	01/98 – 12/04	16	4
				128	11/02 – now	64	4
				128	11/05 – 12/04	32	4
				128	11/05 – 12/04	32	4
H	22	1	256	256	11/04 – now	1024	0

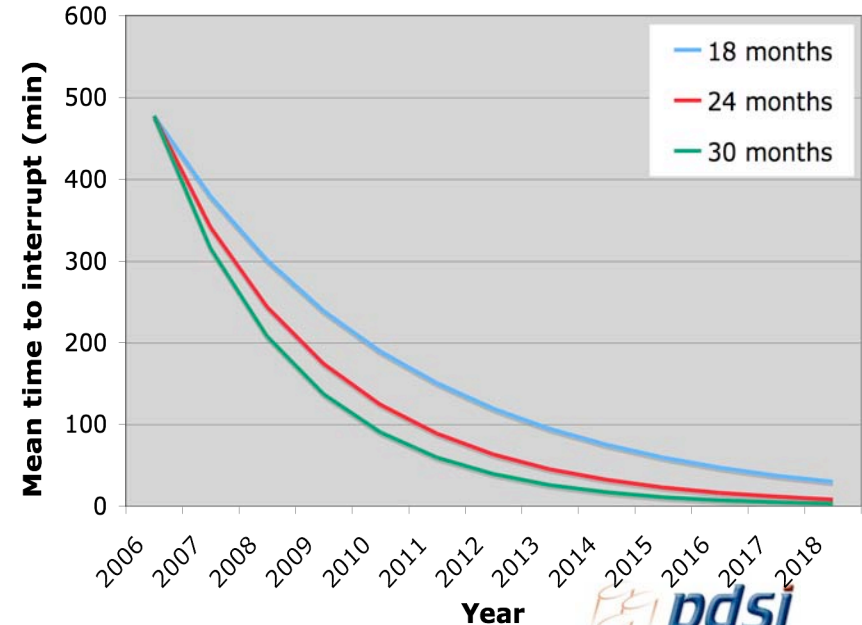
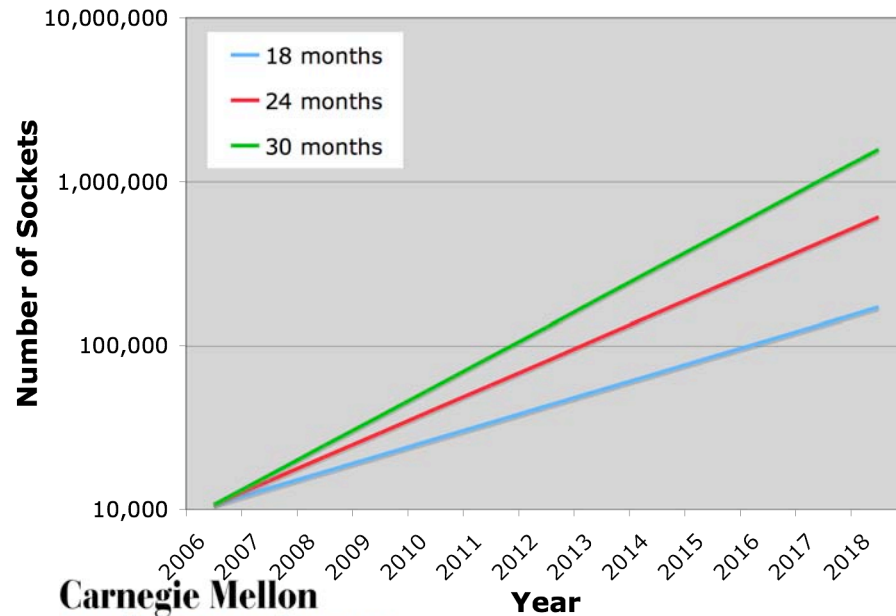
Table 1. Overview of systems. Systems 1–18 are SMP-based, and systems 19–22 are NUMA-based.

Best model: failures track # of processor chips



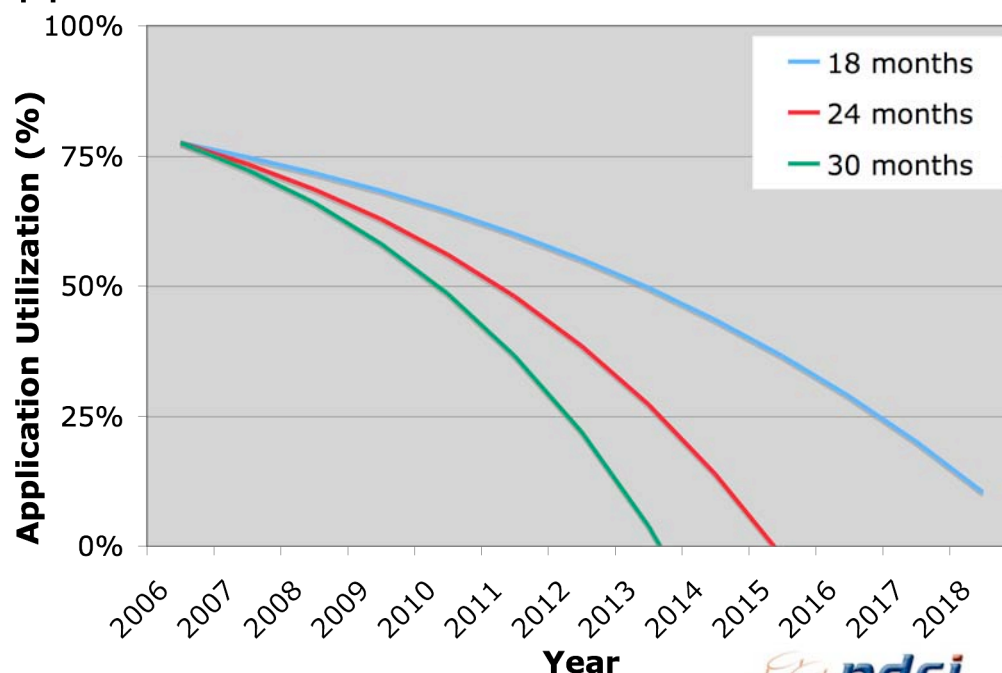
Petascale projections: more failures

- Con't top500.org annual 2X peak FLOPS
 - Set to 1 PF plan for ORNL Baker, LANL Roadrunner in 2008
- Cycle time flat; Cores/chip on Moore's law
 - Consider 2X cores per chip every 18, 24, 30 months
- # sockets, $1/\text{MTTI}$ = failure rate up 25%-50% per year
 - Optimistic 0.1 failures per year per socket (vs. historic 0.25)



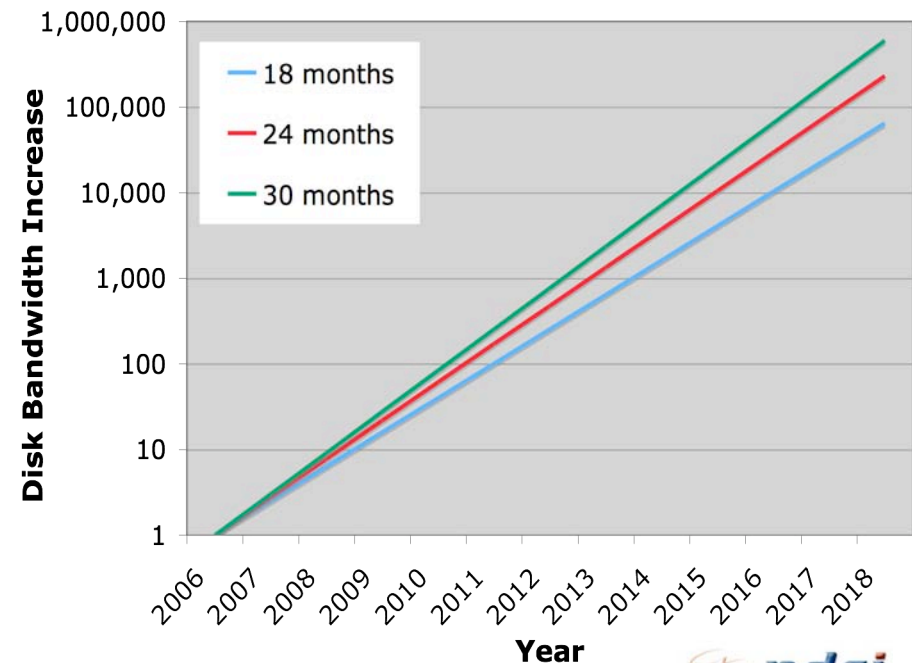
Checkpointing app's utilization

- Periodic (p) app pause to capture checkpoint (t)
- On failure, roll back & restart from checkpoint
- Balanced: Mem, disk speed track FLOPS (constant t)
 - $1 - \text{App util} = t / p + p / (2 * \text{MTTI})$; $p^2 = 2 * t * \text{MTTI}$
 - If MTTI was constant, app utilization would be too
- But MTTI drops
- So Application utilization drops
- Half machine gone soon
- Not acceptable



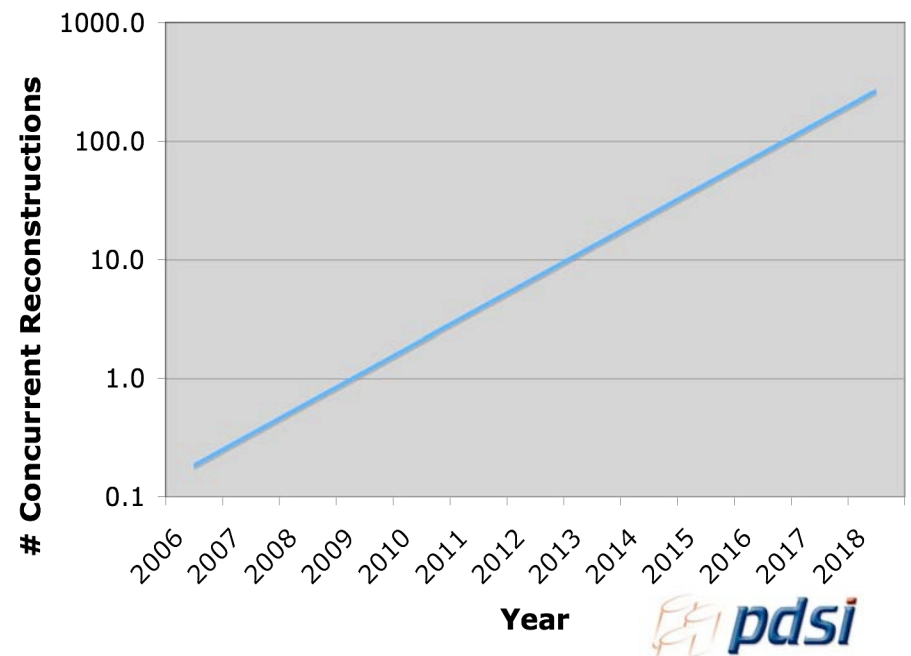
Storage bandwidth to the rescue?

- Increase storage bandwidth to counter for MTTI?
- First, balance means storage bandwidth tracks FLOPS, 2X per year, but disks 20% faster each year
 - Number of disks up 67% each year just for balance
- Doesn't counter MTTI
 - # Disks up 130% / year !
 - Faster than sockets, faster than FLOPS!
 - If system cost grows as # disks vs # sockets
 - Total costs increasingly going into storage (even just for balance)



While on storage issues ...

- Increasing disk bandwidth: more disks & disk failures
 - Data shows 3% per year are replaced [Schroeder, FAST07]
- RAID (level 5, 6 or stronger codes) protect data
 - At cost of online reconstruction of all lost data
 - Larger disks: longer reconstructions, hours become days
- Consider # concurrent reconstructions
- 10-20% now, but
- Soon 100s of concurrent reconstructions
- Storage does not have checkpoint/restart model
- Design normal case for many failures

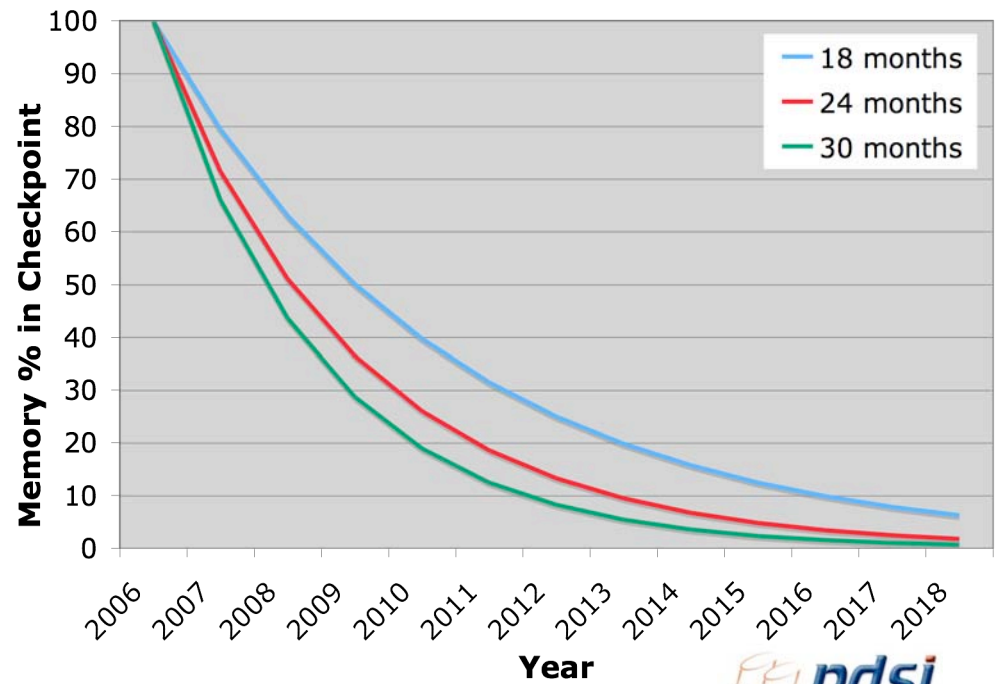


Smaller applications escape

- If an app uses $1/n$ of machine (sockets & memory)
 - $1 - \text{App util} = t/n / p + p / (2 * n * \text{MTTI})$; $p^2 = 2 * t/n * n * \text{MTTI}$
 - Checkpoint overhead of subset resources is reduced by n
 - Assume full storage bandwidth avail for small checkpoint
- If app uses constant resources, it counters MTTI
 - ie., less and less of biggest machine
- Peak machines, when sliced up, see less inefficiency
- But Hero Apps, those that motivate ever bigger machines, gain nothing
 - Hero Apps are primary target of revisiting checkpoint/restart

Applications squeeze checkpoints?

- So far, assumed checkpoint size is memory
- Could Apps counter MTTI with compression?
 - Lots of cycles for compression when saturating storage
- Size of checkpoint has to decrease with MTTI
 - Smaller fraction of memory with each machine
 - Drop 25-50% per year
- If possible
- Cache checkpoint in other node's memory
- Decrease pressure on storage bandwidth and storage costs



Change fault tolerance scheme?

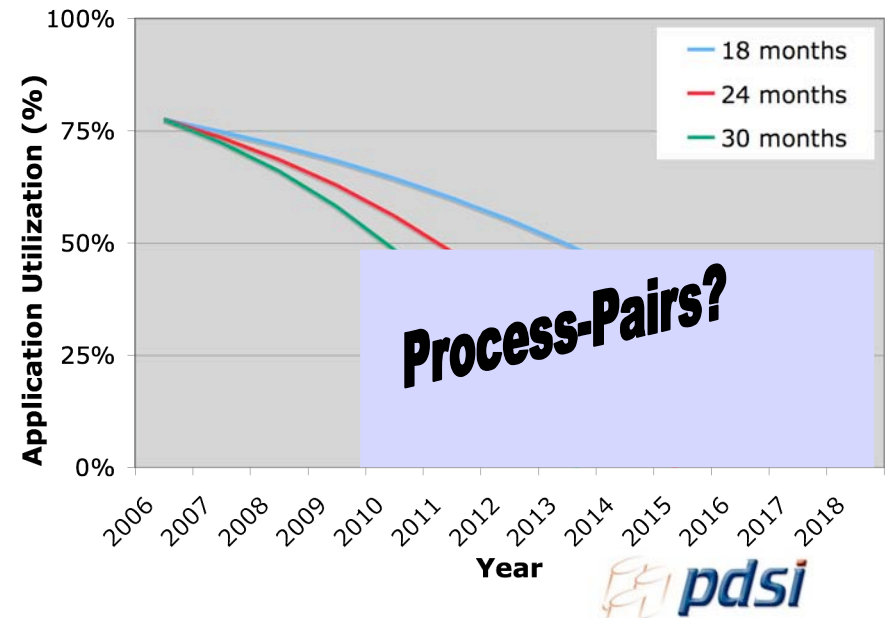
- Classic reliable computing: process-pairs
 - Distributed, parallel simulation as transaction (message) processing
 - Automation possible w/ hypervisors
- Deliver all incoming messages to both
- Match outgoing messages from both
- 50% hardware overhead + slowdown of pair synch
- But if App Utilization is falling under 50% anyway
- No stopping to checkpoint
 - Less pressure on storage bandwidth except for visualization checkpoints

A NonStop* Kernel

Joel F. Bartlett
Tandem Computers Inc.

Abstract © 1981 ACM 0-89791-062-1-12/81-0022

The Tandem NonStop System is a fault-tolerant [1], expandable, and distributed computer system designed expressly for online transaction processing. This paper describes the key primitives of the kernel of the operating system. The first section describes the basic hardware building blocks and introduces their software analogs: processes and messages. Using these primitives, a mechanism that allows fault-tolerant resource access, the process-pair, is described. The paper concludes with some observations on this type of system structure and on actual use of the system.



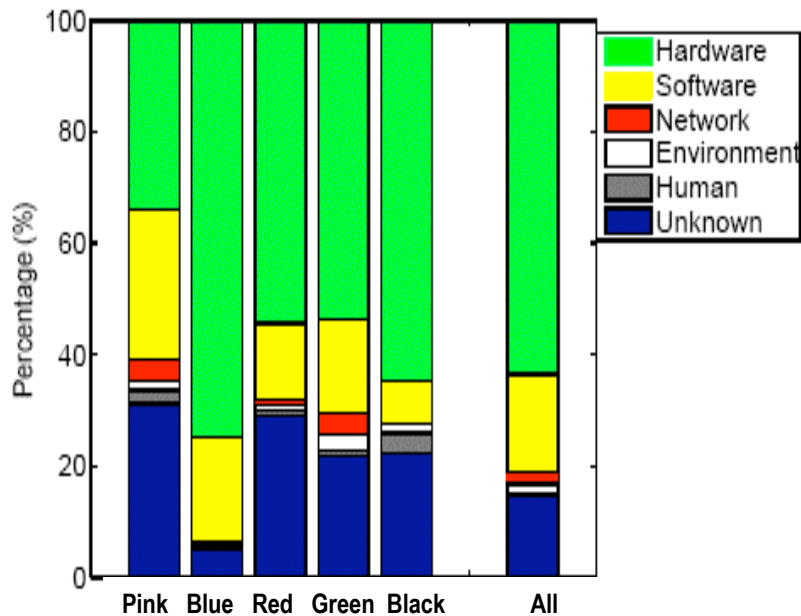
Closing refrain

- Sharing failure data powerful for system research
- Failure rates proportional to number of components
 - Specifically, # sockets in petascale computer (maybe worse)
- If peak compute continues to outstrip Moore's law
 - MTTI will drop, forcing more checkpoints & restarts
 - Effective application utilization will drop significantly
 - Storage bandwidth fixes too expensive (& too hard)
- Hero apps, wanting all the resources, bear burden
 - Small apps don't feel the inefficiency
 - Spending cycles to compress checkpoints good idea
 - When at 50% utilization, consider switch to process-pairs

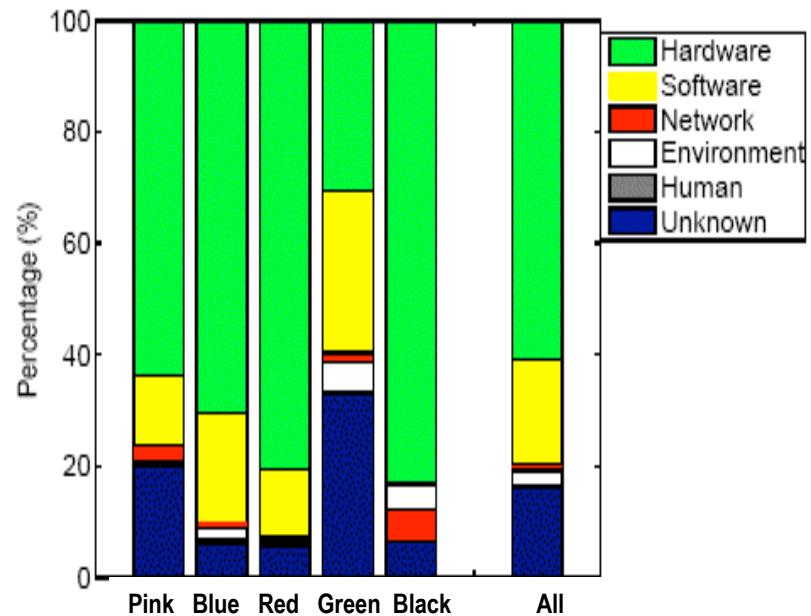
garth@cs.cmu.edu & www.pdsi-scidac.org

Q&A

What is the common root cause of failures?



Relative frequency of root cause by system type.



Fraction of total repair time caused by each root cause.

- Breakdown varies across systems
- Hardware and software most common root cause, and largest contributors to repair times

Relative frequency of disk replacements

The top ten of replaced components

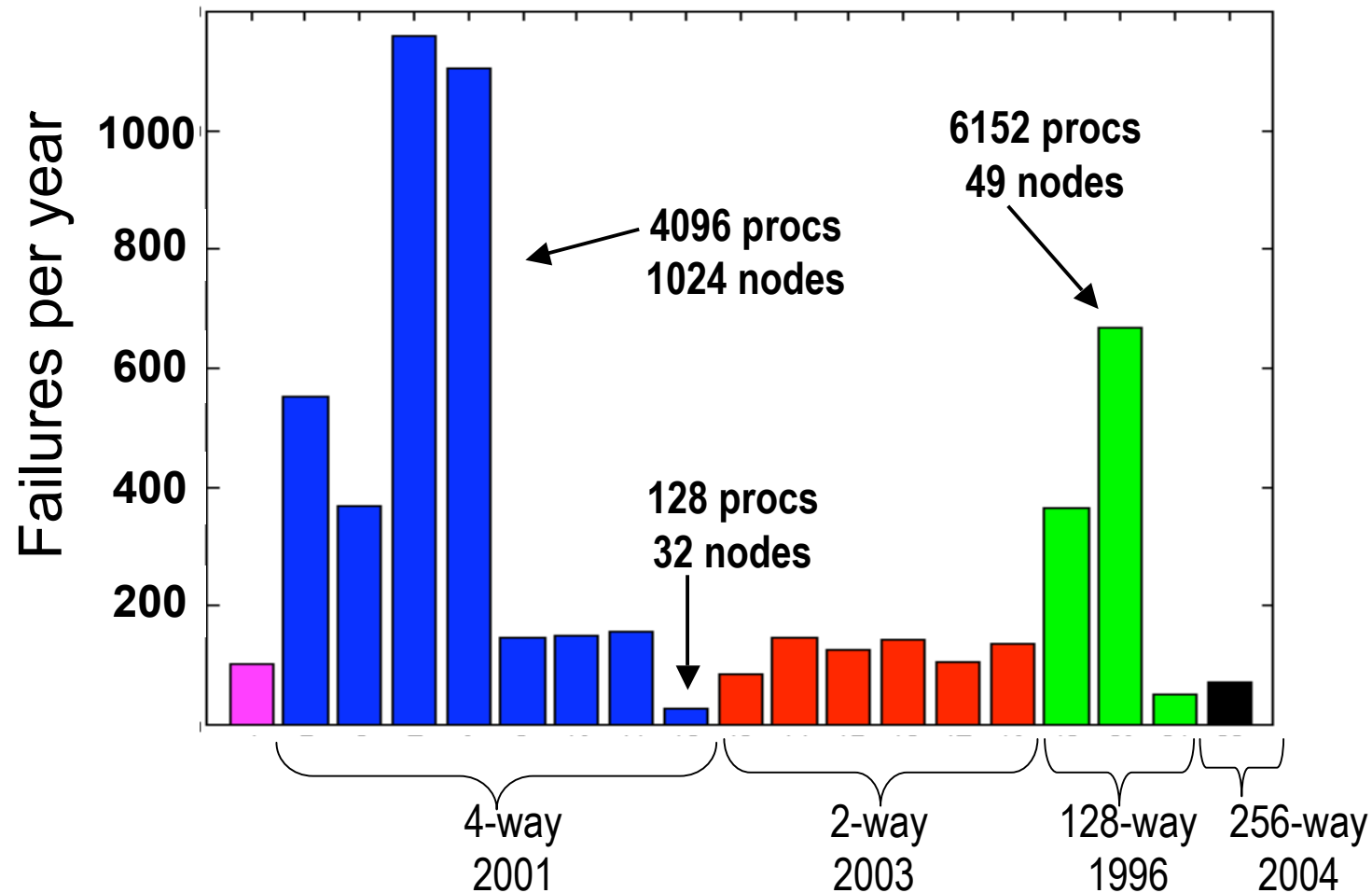
HPC1	
Component	%
Hard drive	30.6
Memory	28.5
Misc/Unk	14.4
CPU	12.4
PCI motherboard	4.9
Controller	2.9
QSW	1.7
Power supply	1.6
MLB	1.0
SCSI BP	0.3

COM1	
Component	%
Power supply	34.8
Memory	20.1
Hard drive	18.1
Case	11.4
Fan	8.0
CPU	2.0
SCSI Board	0.6
NIC Card	1.2
LV Power Board	0.6
CPU heatsink	0.6

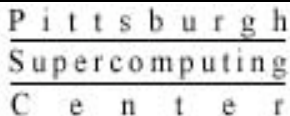




COM2	
Component	%
Hard drive	49.1
Motherboard	23.4
Power supply	10.1
RAID card	4.1
Memory	3.4
SCSI cable	2.2
Fan	2.2
CPU	2.2
CD-ROM	0.6
Raid Controller	0.6

- All hardware fails, though disks failures often common

System failure rate highly variable

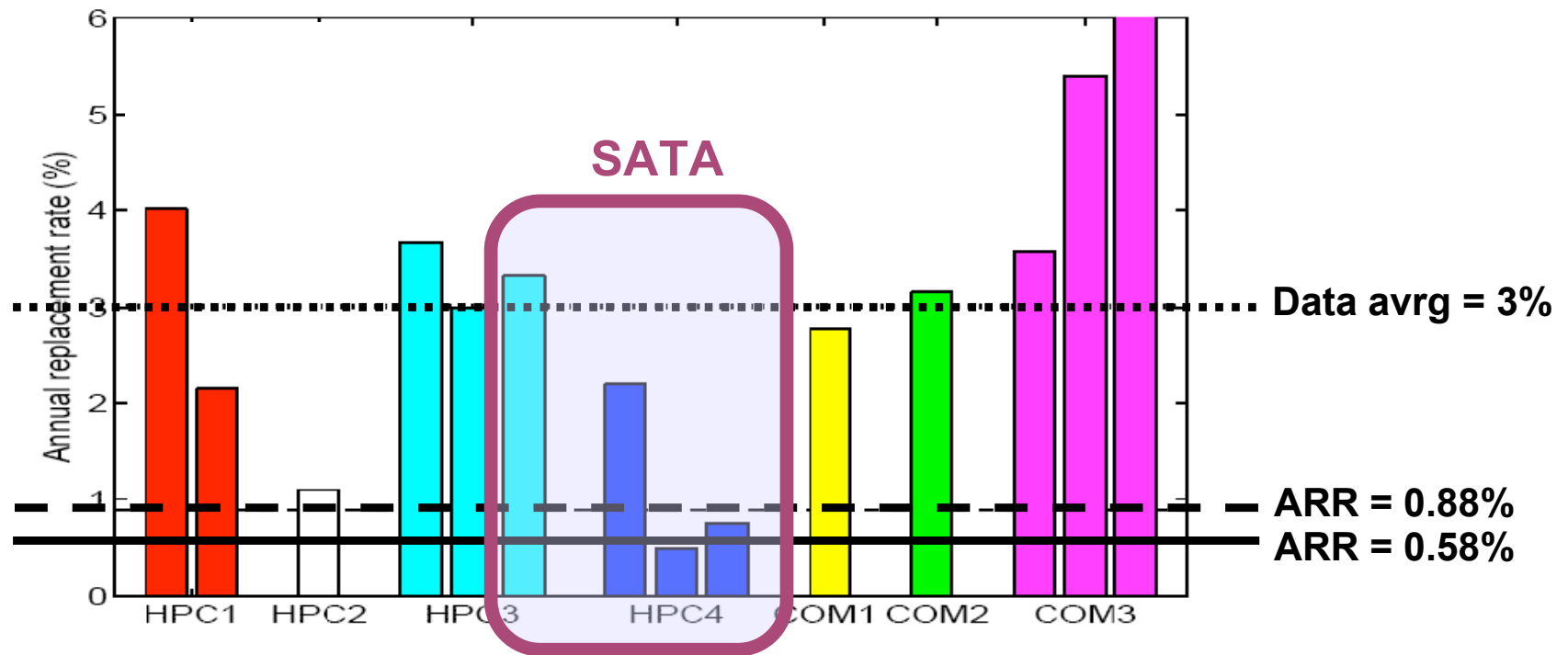


Failure data: hardware replacement logs

		Type of drive	Count	Duration
	HPC1	18GB 10K RPM SCSI 36GB 10K RPM SCSI	3,400	5 yrs
	HPC2	36GB 10K RPM SCSI	520	2.5 yrs
 Supercomputing X	HPC3	15K RPM SCSI 15K RPM SCSI 7.2K RPM SATA	14,208	1 yr
 Various HPCs	HPC4	250GB SATA 500GB SATA 400GB SATA	13,634	3 yrs
 Internet services Y	COM1	10K RPM SCSI	26,734	1 month
	COM2	15K RPM SCSI	39,039	1.5 yrs
	COM3	10K RPM FC-AL 10K RPM FC-AL 10K RPM FC-AL 10K RPM FC-AL	3,700	1 yr

Annual disk replacement rate (ARR)

- Datasheet MTTFs are 1,000,000 to 1,500,000 hours.
- => Expected annual replacement rate (ARR): 0.58 - 0.88 %.



- Poor evidence for SATA fail rates higher than SCSI or FC